

Part 1:

MEMBUAT PETA ONLINE DENGAN LEAFLETJS


Tentang LeafletJS

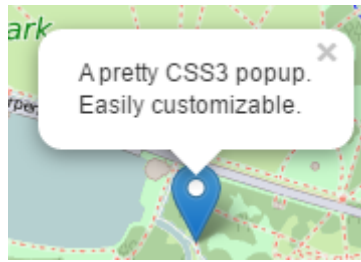
LeafletJS merupakan library atau kumpulan fungsi berbasis Javascript yang digunakan untuk menampilkan peta interaktif pada halaman web. Leaflet menyediakan Map API (Application Programming Interface) yang memudahkan web developer untuk menampilkan peta berbasis Tile pada halaman web. Pengguna peta juga dapat berinteraksi dengan menggunakan fungsi telah disediakan oleh Leaflet.

Dibandingkan Map API lainnya seperti OpenLayer, GMAP API, ArcGIS JS API atau Mapbox API, Leaflet memiliki keunggulan dalam kecepatan akses dan ukuran berkas yang kecil. Meskipun memiliki ukuran berkas yang lebih kecil, LeafletJS memiliki fungsi web mapping yang cukup lengkap. Fungsionalitas LeafletJS ini dapat ditingkatkan dengan menggunakan berbagai Plugin yang tersedia secara cuma-cuma.

Komponen LeafletJS

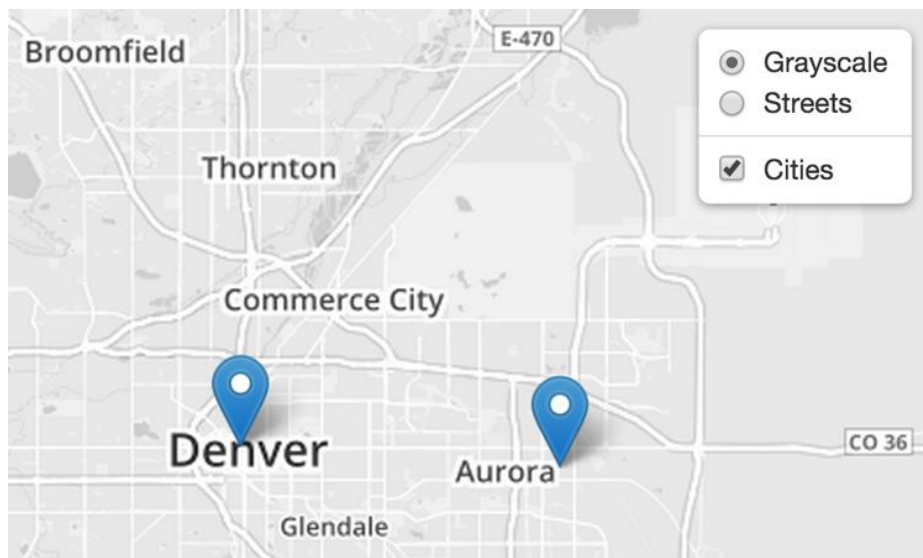
LeafletJS, sebagaimana juga webmap API lainnya, memiliki beberapa komponen dasar sebagai berikut:

- **Map.** Adalah komponen induk yang memuat berbagai komponen lainnya. Bayangkan komponen Map sebagai muka peta kosong yang nantinya akan dapat diisi dengan komponen lain (seperti tilelayer, marker, dan lain sebagainya). Pada komponen inilah didefinisikan ukuran peta pada halaman web (melalui fungsi CSS, *width* dan *height*), koordinat pusat (*center*) peta (dalam latitude dan longitude), serta level zoom awal (antara 0-20, level 20 menunjukkan perbesaran paling tinggi). Komponen dapat ditambahkan pada Map melalui method **.addTo(Map)**.
- **Tilelayer** (seringkali disebut dengan “slippy map”), merupakan komponen yang menyediakan latar belakang peta pada sebuah webmap/peta online. Peta latar yang disediakan ditampilkan dalam bentuk kotak-kotak (‘tile’) yang memiliki tampilan berbeda pada level zoom yang berbeda. LeafletJS menyediakan fungsi (**L.Tilelayer**) untuk memanggil layer peta latar belakang dari berbagai sumber, seperti Openstreetmap (OSM), Mapbox, ESRI, dan lain sebagainya. Layer dasar yang dipanggil dari Tilelayer ini disebut juga sebagai **Base Map**
- **Marker.** Simbologi sederhana untuk menyatakan titik. Default simbologi untuk marker pada Leaflet adalah , meskipun LeafletJS dan Map API lainnya memungkinkan developer peta untuk mengganti simbologi dengan bebas. Fungsi pada Leaflet untuk memunculkan marker adalah **L.Marker**. Marker digunakan untuk memberi penjelasan pada data lokasi berupa titik.
- **Popup.** Adalah jendela kecil berisi informasi yang terikat dengan marker tertentu. Popup biasanya digunakan untuk menunjukkan informasi terkait titik tersebut, misalnya berupa rangkaian teks, gambar atau grafik lain. Popup dipanggil dengan menggunakan method **bindPopup** pada objek L.Marker (**L.Marker.bindPopup**).



Contoh popup yang terikat pada pada sebuah marker

- **Event.** Event merupakan ‘kejadian’ yang dapat diamati oleh Leaflet pada muka peta. Fungsi Event digunakan untuk menyediakan interaktifitas dengan pengguna. Misalnya, gerakan mouse pengguna merupakan Event yang dapat diterjemahkan menjadi tampilan koordinat di mana lokasi pointer mouse berada. Contoh lain: Klik pada marker merupakan event yang dapat digunakan untuk memicu (‘trigger’) munculnya popup atau fungsi lain. Fungsi event terikat dengan objek-objek pada Leaflet, sehingga tidak membutuhkan kode tertentu untuk dipanggil.
- **Control.** Kontrol peta pada Leaflet merupakan pelengkap muka peta. **L.Control** dalam Leaflet merupakan tombol Zoom pada peta, menu pencarian, menu pemilihan layer, menu penggambaran dan seterusnya.



Sebuah ‘Control’ pada muka peta Leaflet yang menyediakan fungsi pemilihan layer

- **Vector Layer.** Layer pada Leaflet merupakan data spasial jenis Vector yang dapat ditambahkan pada komponen Map Leaflet. LeafletJS memiliki fungsi penggambaran layer tersendiri, misalnya dengan menggunakan fungsi **L.Polygon**. Layer dari sumber lain (misalnya file SHP) juga dapat ditambahkan dengan melakukan konversi terlebih dahulu menjadi format GeoJSON (misalnya dengan menggunakan QGIS), kemudian memanggil layer tersebut menggunakan fungsi **L.GeoJSON** pada LeafletJS.
- **Plugin.** Komponen plugin memperkaya fungsi-fungsi yang sudah ada pada Leaflet dengan berbagai fungsi tambahan yang dapat digunakan apabila dibutuhkan. Berikut adalah beberapa contoh Plugin yang dapat digunakan pada LeafletJS:

Contoh beberapa plugin Leaflet

NAMA PLUGIN	FUNGSI
Leaflet Providers	Menambahkan berbagai Tilelayer dari sumber yang beragam (lihat https://leaflet-extras.github.io/leaflet-providers/preview/)
ESRI Leaflet	Menyediakan fungsi-fungsi pada ArcGIS Javascript API untuk digunakan pada LeafletJS.
Leaflet Awesome Marker	Menampilkan icon yang bervariasi untuk marker Leaflet. Icon dapat dipanggil berdasarkan pada Font Awesome dan Ionicons
Leaflet Photo	Untuk menampilkan foto/gambar yang memiliki Geotag pada Leaflet
Leaflet Marker Cluster	Visualisasi pengelompokan marker berdasarkan lokasi
Leaflet.Shapefile	Fungsi untuk secara langsung memanggil file bertipe SHP untuk ditampilkan pada Leaflet
Leaflet.WFST	Plugin untuk memanggil layanan WFS(T)
Leaflet.Realtime	Plugin untuk menampilkan data secara realtime (misalnya: data tracking GPS)
Leaflet.Control.SideBySide	Untuk menyediakan Control tambahan yang berfungsi untuk menampilkan dua buah peta secara berdampingan

Daftar lengkap dari beberapa marker Leaflet tersedia pada alamat resmi untuk penyimpanan plugin Leaflet: <http://leafletjs.com/plugins.html>

- **Services.** Merupakan data tambahan yang dapat dimuat pada muka peta Leaflet. Data tambahan yang dimaksud dapat berupa layanan WMS (Web Map Services) yang bersumber dari server data spasial (misalnya Geoserver). Layer Services pada Leaflet dikategorikan sebagai Tilelayer, dan dapat dipanggil dengan menggunakan fungsi **L.Tilelayer.WMS**. Layanan lain semisal WFS dapat ditambahkan dengan menggunakan plugin, karena Leaflet hanya bertujuan untuk menampilkan fungsi peta sederhana seringkali mungkin. Adapun untuk keperluan yang lebih kompleks dapat digunakan ArcGIS JS API atau Openlayers (OL).

Komponen LeafletJS yang telah disebutkan memiliki kesamaan dengan hampir semua webmap API lainnya. Dengan mempelajari Leaflet yang secara struktur lebih sederhana dibandingkan dengan webmap API lain akan memudahkan dalam memahami webmap API lain yang lebih kompleks.

Membuat Peta dengan LeafletJS

Pada bagian ini akan dijelaskan mengenai langkah-langkah pembuatan halaman peta berbasis web dengan menggunakan LeafletJS. Untuk dapat memahami dengan baik bagian ini, sangat disarankan untuk mempelajari terlebih dahulu mengenai HTML, CSS dan Javascript

1. Membuat kerangka dokumen HTML

Pertama-tama, buat sebuah dokumen HTML (misalnya dengan menggunakan Notepad++) dengan isi sebagai berikut:

```
<!DOCTYPE html>
<html>
<head>
  <title>Leaflet Web Map</title>
  <style>
  </style>
</head>
<body>
  <script>
  </script>
</body>
</html>
```

Berikan nama pada halaman tersebut. Misalnya **petaleaflet.html**. Kerangka ini kemudian akan diisi dengan komponen-komponen Leaflet sebagaimana disebutkan di atas.

2. Memanggil Library LeafletJS

LeafletJS merupakan sebuah Library Javascript yang terdiri dari komponen tampilan (css) dan komponen logis (javascript). Sebagaimana pada bagian Javascript Library lainnya, LeafletJS dapat dipanggil dengan dua cara:

- Mengunduh file yang berisi library LeafletJS (leaflet.css dan leaflet.js) dari <http://leafletjs.com/download.html>
- Menggunakan CDN (Content Delivery Network) yang menyimpan kode Leaflet

Kode Leaflet yang tersimpan pada CDN dapat diakses dengan cara berikut:

```
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
```

Tempatkan kedua baris kode ini pada bagian <head> pada dokumen HTML yang sudah dibuat.

```
<!DOCTYPE html>
<html>
<head>
  <title>Leaflet Web Map</title>
  <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
  <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
  <style>
  </style>
</head>
<body>
  <script>
  </script>
</body>
</html>
```

3. Menambahkan komponen 'Map'

Tambahkan komponen utama peta (Map) dengan menggunakan tag HTML 'div' pada bagian <body>.

```
<body>
  <div id="map"></div>
  <script>
  </script>
</body>
```

Selanjutnya, tentukan lebar dan tinggi muka peta yang akan dibuat dengan menggunakan kode css yang ditempatkan pada bagian <style></style> pada <head>.

```
<style>
  #map {
    width: 960px;
    height: 500px;
  }
</style>
```

Dengan demikian, komponen utama kerangka peta telah selesai dibuat.

4. Pengaturan dasar Map dan Memuat Basemap

Langkah selanjutnya adalah mengatur pusat peta (center) dan tingkat perbesaran. Pada bagian <script> di <body>, masukkan kode berikut:

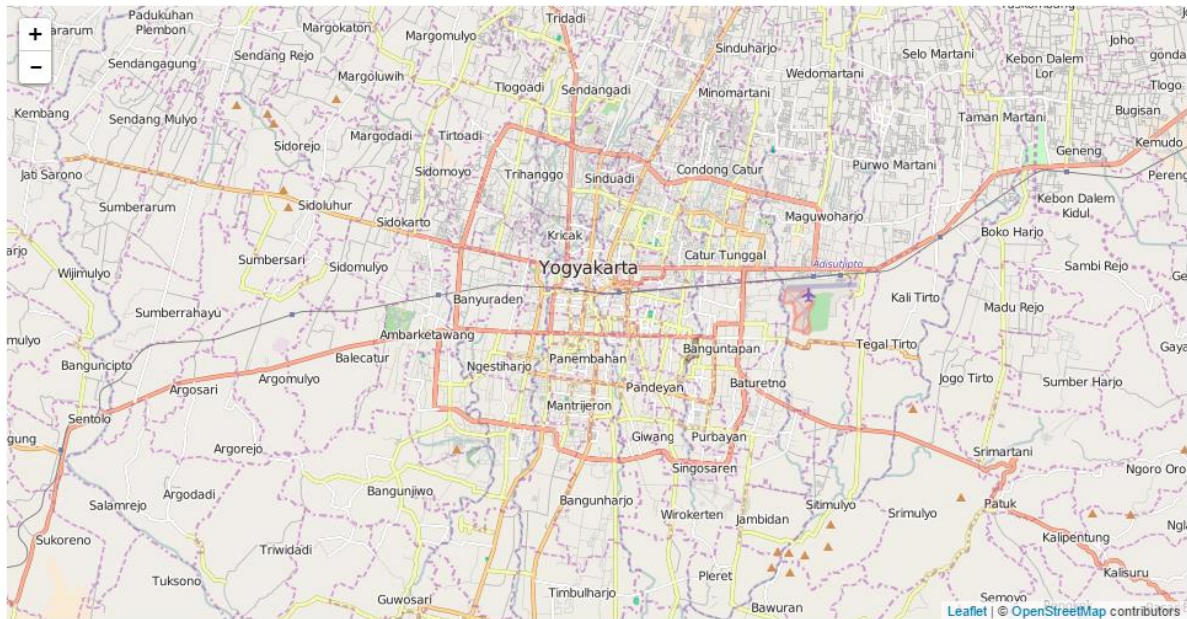
```
var map = L.map('map', {
  center: [-7.79558, 110.36949],
  zoom: 12
});
```

Pengaturan tersebut akan memuat peta dengan pusat peta pada lokasi yang diberikan (sekitar Jogja) dengan tingkat perbesaran 12.

Masih pada tag <script> yang sama, masukkan kode berikut untuk memuat Basemap berupa Tilelayer dari OpenStreetMap:

```
L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
}).addTo(map);
```

Saat ini peta sudah dapat ditampilkan. Apabila file petaleaflet.html dipanggil pada browser, maka peta berikut akan muncul:



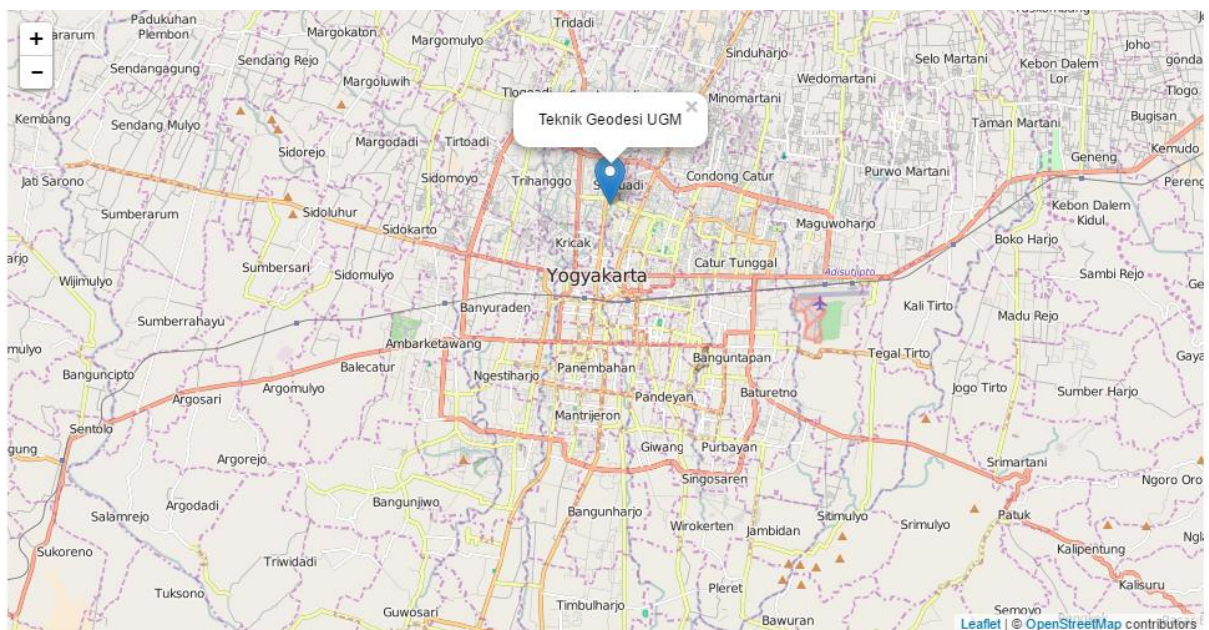
Peta Leaflet dengan komponen paling sederhana

5. Menambahkan marker dan popup

Untuk menambahkan marker dan popup, gunakan script berikut pada lokasi yang sama (tag <script>):

```
var marker = L.marker([-7.7639847, 110.3704093]).bindPopup('Teknik Geodesi UGM').addTo(map);
```

Sekarang peta yang dihasilkan adalah sebagai berikut:



Peta Leaflet dengan marker dan popup

Rujukan

Rujukan untuk belajar LeafletJS mandiri banyak tersedia di dunia maya. Berikut diantaranya:

- <http://leafletjs.com/>
- <http://maptimeboston.github.io/leaflet-intro/>
- <https://leanpub.com/leaflet-tips-and-tricks/read>
- <http://joshuafrazier.info/leaflet-basics/>

Sumber icon yang dapat digunakan untuk marker Leaflet

- Glyphicons (<http://glyphicons.com/>)
- Noun Project (<https://thenounproject.com/>)
- Fontawesome (<http://fontawesome.github.io/Font-Awesome/>)
- Ionic Icons (<http://ionicons.com/>)
- Maki (<https://www.mapbox.com/maki/>)